

LAPORAN PRAKTIKUM
MODUL 6
“DOUBLE LINKED LIST (BAGIAN PERTAMA)”



Disusun oleh:

Wendri Tri Pambudi (21104048 / SE07-01)

Dosen:

Yudha Islami Sulistya, S.Kom., M.Cs

PROGRAM STUDI REKAYASA PERANGKAT LUNAK

FAKULTAS INFORMATIKA

UNIVERSITAS TELKOM PURWOKERTO

2024

Tugas Pendahuluan

1. Soal 1: Menambahkan Elemen di Awal dan Akhir DLL

Deskripsi Soal: Buatlah program yang mengizinkan pengguna menambahkan elemen ke dalam Doubly Linked List di awal dan di akhir list.

Instruksi

1. Implementasikan fungsi `insertFirst` untuk menambahkan elemen di awal list.
2. Implementasikan fungsi `insertLast` untuk menambahkan elemen di akhir list.
3. Tampilkan seluruh elemen dalam list dari depan ke belakang setelah penambahan dilakukan.

Code Soal 1:

```
06_Double_Linked_List_Bagian_1 > TP > C++ SOAL_01.cpp > printList(Node *)
1  #include <iostream>
2  using namespace std;
3
4  struct Node {
5      int data;
6      Node* prev;
7      Node* next;
8  };
9
10 void insertFirst(Node* &head, int value) {
11     Node* newNode = new Node();
12     newNode->data = value;
13     newNode->prev = NULL;
14     newNode->next = head;
15
16     if (head != NULL) {
17         head->prev = newNode;
18     }
19     head = newNode;
20 }
21
22 void insertLast(Node* &head, int value) {
23     Node* newNode = new Node();
24     newNode->data = value;
25     newNode->next = NULL;
26
27     if (head == NULL) {
28         newNode->prev = NULL;
29         head = newNode;
30         return;
31     }
32
33     Node* temp = head;
34     while (temp->next != NULL) {
35         temp = temp->next;
36     }
37
38     temp->next = newNode;
39     newNode->prev = temp;
40 }
41
```

```

06_Double_Linked_List_Bagian_1 > TP > C++ SOAL_01.cpp > insertLast(Node *8, int)
41
42 void printList(Node* head) {
43     Node* temp = head;
44     cout << "DAFTAR ANGGOTA LIST: ";
45     while (temp != NULL) {
46         cout << temp->data;
47         if (temp->next != NULL) {
48             cout << " <-> ";
49         }
50         temp = temp->next;
51     }
52     cout << endl;
53 }
54
55 int main() {
56     Node* head = NULL;
57
58     int firstElement;
59     cout << "Masukkan elemen pertama: ";
60     cin >> firstElement;
61     insertLast(head, firstElement);
62     printList(head);
63
64     int secondElement;
65     cout << "Masukkan elemen kedua di awal: ";
66     cin >> secondElement;
67     insertFirst(head, secondElement);
68     printList(head);
69
70     int thirdElement;
71     cout << "Masukkan elemen ketiga di akhir: ";
72     cin >> thirdElement;
73     insertLast(head, thirdElement);
74     printList(head);
75
76     return 0;
77 }
78

```

Ouput Soal 1:

```

Masukkan elemen pertama: 10
DAFTAR ANGGOTA LIST: 10
Masukkan elemen kedua di awal: 5
DAFTAR ANGGOTA LIST: 5 <-> 10
Masukkan elemen ketiga di akhir: 20
DAFTAR ANGGOTA LIST: 5 <-> 10 <-> 20

```

2. Soal 2: Menghapus Elemen di Awal dan Akhir DLL

Deskripsi Soal: Buatlah program yang memungkinkan pengguna untuk menghapus elemen pertama dan elemen terakhir dalam Doubly Linked List.

Instruksi

1. Implementasikan fungsi `deleteFirst` untuk menghapus elemen pertama.
2. Implementasikan fungsi `deleteLast` untuk menghapus elemen terakhir.
3. Tampilkan seluruh elemen dalam list setelah penghapusan dilakukan.

Code Soal 2:

06_Double_Linked_List_Bagian_1 > TP > C++ SOAL_02.cpp > deleteFirst(Node *&h)

```
4 struct Node {
5     int data;
6     Node* prev;
7     Node* next;
8 };
9
10 void insertLast(Node* &head, int value) {
11     Node* newNode = new Node();
12     newNode->data = value;
13     newNode->next = NULL;
14
15     if (head == NULL) {
16         newNode->prev = NULL;
17         head = newNode;
18         return;
19     }
20     Node* temp = head;
21     while (temp->next != NULL) {
22         temp = temp->next;
23     }
24     temp->next = newNode;
25     newNode->prev = temp;
26 }
27
28 void deleteFirst(Node* &head) {
29     if (head == NULL) {
30         cout << "List kosong, tidak bisa menghapus." << endl;
31         return;
32     }
33     Node* temp = head;
34     head = head->next;
35     if (head != NULL) {
36         head->prev = NULL;
37     }
38     delete temp;
39 }
```

06_Double_Linked_List_Bagian_1 > TP > C++ SOAL_02.cpp > ...

```
40
41 void deleteLast(Node* &head) {
42     if (head == NULL) {
43         cout << "List kosong, tidak bisa menghapus." << endl;
44         return;
45     }
46
47     if (head->next == NULL) {
48         delete head;
49         head = NULL;
50         return;
51     }
52
53     Node* temp = head;
54     while (temp->next != NULL) {
55         temp = temp->next;
56     }
57
58     temp->prev->next = NULL;
59     delete temp;
60 }
61
62 void printList(Node* head) {
63     Node* temp = head;
64     cout << "DAFTAR ANGGOTA LIST: ";
65     while (temp != NULL) {
66         cout << temp->data;
67         if (temp->next != NULL) {
68             cout << " <-> ";
69         }
70         temp = temp->next;
71     }
72     cout << endl;
73 }
```

```

06_Double_Linked_List_Bagian_1 > TP > C++ SOAL_02.cpp > main()
75  int main() {
76      Node* head = NULL;
77
78      int firstElement;
79      cout << "Masukkan elemen pertama: ";
80      cin >> firstElement;
81      insertLast(head, firstElement);
82      printList(head);
83
84      int secondElement;
85      cout << "Masukkan elemen kedua di akhir: ";
86      cin >> secondElement;
87      insertLast(head, secondElement);
88      printList(head);
89
90      int thirdElement;
91      cout << "Masukkan elemen ketiga di akhir: ";
92      cin >> thirdElement;
93      insertLast(head, thirdElement);
94      printList(head);
95      cout << "======" << endl;
96      cout << "Menghapus elemen pertama dan terakhir:" << endl;
97      deleteFirst(head);
98      deleteLast(head);
99      cout << "List setelah dihapus: ";
100     printList(head);
101
102     return 0;
103 }
104

```

Ouput Soal 2:

```

Masukkan elemen pertama: 10
DAFTAR ANGGOTA LIST: 10
Masukkan elemen kedua di akhir: 15
DAFTAR ANGGOTA LIST: 10 <-> 15
Masukkan elemen ketiga di akhir: 20
DAFTAR ANGGOTA LIST: 10 <-> 15 <-> 20
=====
Menghapus elemen pertama dan terakhir:
List setelah dihapus: DAFTAR ANGGOTA LIST: 15

```

3. Soal 3: Menampilkan Elemen dari Depan ke Belakang dan Sebaliknya

Deskripsi Soal: Buatlah program yang memungkinkan pengguna memasukkan beberapa elemen ke dalam Doubly Linked List. Setelah elemen dimasukkan, tampilkan seluruh elemen dalam list dari depan ke belakang, kemudian dari belakang ke depan.

Instruksi

1. Implementasikan fungsi untuk menampilkan elemen dari depan ke belakang.
2. Implementasikan fungsi untuk menampilkan elemen dari belakang ke depan.
3. Tambahkan 4 elemen ke dalam list dan tampilkan elemen tersebut dalam dua arah.

Code Soal 3:

```
06_Double_Linked_List_Bagian_1 > TP > C++ SOAL_03.cpp > printListForward(Node *)
1  #include <iostream>
2  using namespace std;
3
4  struct Node {
5      int data;
6      Node* prev;
7      Node* next;
8  };
9
10 void insertLast(Node* &head, int value) {
11     Node* newNode = new Node();
12     newNode->data = value;
13     newNode->next = NULL;
14
15     if (head == NULL) {
16         newNode->prev = NULL;
17         head = newNode;
18         return;
19     }
20
21     Node* temp = head;
22     while (temp->next != NULL) {
23         temp = temp->next;
24     }
25
26     temp->next = newNode;
27     newNode->prev = temp;
28 }
29
30 void printListForward(Node* head) {
31     Node* temp = head;
32     cout << "Daftar elemen dari depan ke belakang: ";
33     while (temp != NULL) {
34         cout << temp->data;
35         if (temp->next != NULL) {
36             cout << " <-> ";
37         }
38         temp = temp->next;
39     }
40     cout << endl;
41 }
42
```

```

06_Double_Linked_List_Bagian_1 > TP > C++ SOAL_03.cpp > main()
43 void printListBackward(Node* head) {
44     if (head == NULL) return;
45
46     Node* temp = head;
47     while (temp->next != NULL) {
48         temp = temp->next;
49     }
50
51     cout << "Daftar elemen dari belakang ke depan: ";
52     while (temp != NULL) {
53         cout << temp->data;
54         if (temp->prev != NULL) {
55             cout << " <-> ";
56         }
57         temp = temp->prev;
58     }
59     cout << endl;
60 }
61
62 int main() {
63     Node* head = NULL;
64
65     int value;
66     for (int i = 1; i <= 4; i++) {
67         cout << "Masukkan elemen ke-" << i << ": ";
68         cin >> value;
69         insertLast(head, value);
70     }
71
72     printListForward(head);
73     printListBackward(head);
74
75     return 0;
76 }

```

Ouput Soal 3:

```

Masukkan elemen ke-1: 1
Masukkan elemen ke-2: 2
Masukkan elemen ke-3: 3
Masukkan elemen ke-4: 4
Daftar elemen dari depan ke belakang: 1 <-> 2 <-> 3 <-> 4
Daftar elemen dari belakang ke depan: 4 <-> 3 <-> 2 <-> 1

```